

MÓDULO 6:

PARADIGMA PROGRAMACION

ORIENTACION OBJETOS

Eduard Lara

INDICE

1. POO Definición
2. Ejemplo entidad Pelota
3. Ejemplo clase Persona
4. Ejemplo Navbar
5. Ejemplo Header alineado
6. Ejemplo Header alineado con constructor
7. Ejemplo Tabla
8. Ejemplo Pagina
9. Ejemplo “Objetos a un Menu”
10. Ejemplo Header Configurado

1. POO: DEFINICION

- La programació orientada a objectes és un paradigma de la programació que descomposa el codi d'una **classe** en unitats anomenades **objectes**.
- Un **objecte** en PHP és una entitat existent en la memòria d'un computador que té unes **propietats** anomenades **atributs** i unes **funcions associades** anomenades **mètodes**.
- Aquests **objectes** venen de **clases**. En realitat, fins ara hem utilitzat una classe, només tenia un únic mètode implícit executable.

1. POO: DEFINICION

- Aquest paradigma és senzill d'entendre però la seva profunditat i efecte alhora de programar és ENORME.
- Bàsicament perquè un objecte en aquest context pot ser QUALSEVOL representació d'una entitat en el món real.
- I és per això que la presa de decisions alhora de generar un o més objectes per resoldre un problema concret del món real serà DECISIVA ja que marcarà l'estructura(inclús l'arquitectura) del programa que fem.

2. EJEMPLO ENTIDAD PELOTA

Objeto: instancia de una clase



Atributos



- Color
- Volumen
- Dureza
- Peso

Métodos

- Rodar



- Rebotar

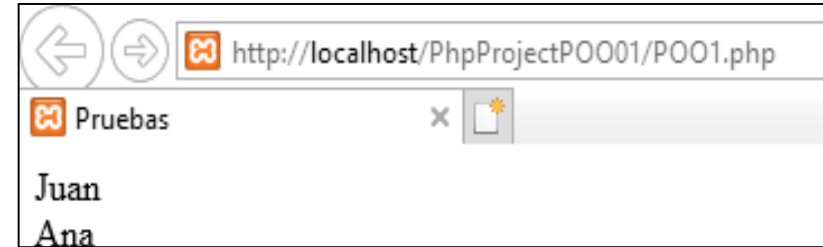


Un objecte(instància d'una classe) seria una representació **concreta** d'una entitat **abstracta**.

De la classe **pilota** tenim un objecte(instància) que representa una pilota concreta.

2. EJEMPLO ENTIDAD PELOTA

```
4 <html>
5 <head>
6 <title>Pruebas</title>
7 </head>
8 <body>
9 <?php
10 class Persona {
11
12     //Atributos de clase
13     private $nombre;
14
15     //Metodos de clase
16     public function inicializar($nom)
17     {
18         $this->nombre=$nom;
19     }
20     public function imprimir()
21     {
22         echo $this->nombre;
23         echo '<br>';
24     }
25 }
26 //Creacion y asignacion de objetos
27 $per1=new Persona();
28 $per1->inicializar('Juan');
29 $per1->imprimir();
30 $per2=new Persona();
31 $per2->inicializar('Ana');
32 $per2->imprimir();
33 ?>
34 </body>
35 </html>
```



Aquí creem una classe que representa informació relativa a una Persona. Tenim un atribut **nombre** i uns mètodes per inicialitzar l'objecte i per imprimir el nom.

Instanciem (creem un objecte) de la classe Persona

La inicialitzem, en aquest cas, li donem nom.

Imprimim el nom d'aquesta persona concreta

Mateix procés però amb diferent persona

3. EJEMPLO CLASE PERSONA

```
9  <?php
10  class Persona {
11
12      //Atributos de clase
13      private $nombre;
14
15      //Metodos de clase
16      public function inicializar($nom)
17      {
18          $this->nombre=$nom;
19      }
20      public function imprimir()
21      {
22          echo $this->nombre;
23          echo '<br>';
24      }
25  }
```

Aquesta és “només” una idea de classe persona. Les implementacions dependran de les necessitats del software

3. EJEMPLO CLASE PERSONA

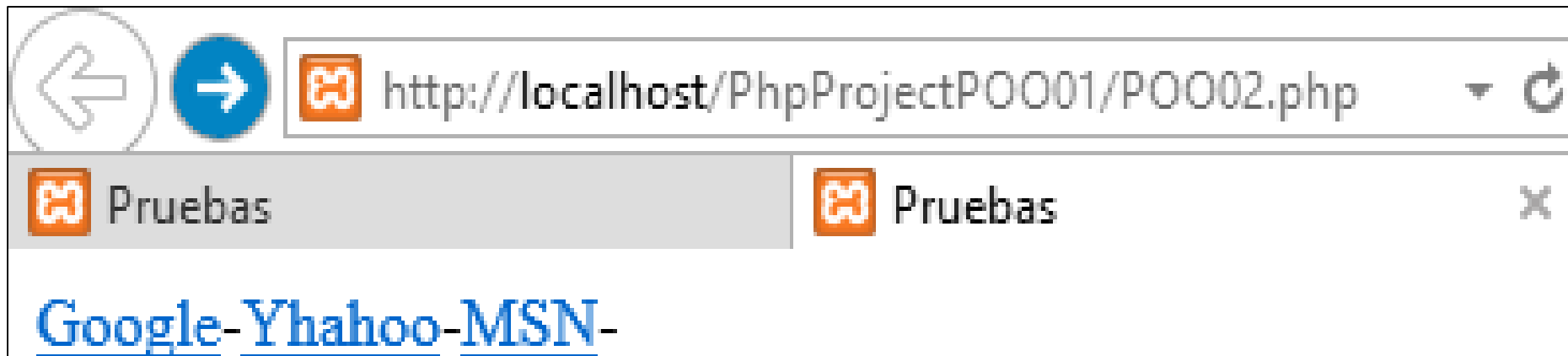
Instanciando la clase persona

```
26      //Creacion y asignacion de objetos
27      $perl=new Persona();
28      $perl->inicializar('Juan');
29      $perl->imprimir();
30      $per2=new Persona();
31      $per2->inicializar('Ana');
32      $per2->imprimir();
33      ?>
34      </body>
35      </html>
```

Aquí fem ús de la implementació de la classe Persona per a fer una representació de dues persones concretes.

4. EJEMPLO “NAVBAR”

Implementar una classe que mostri una llista d'hipervincles en horitzontal(bàsicament un menú d'opcions)



4. EJEMPLO “NAVBAR”

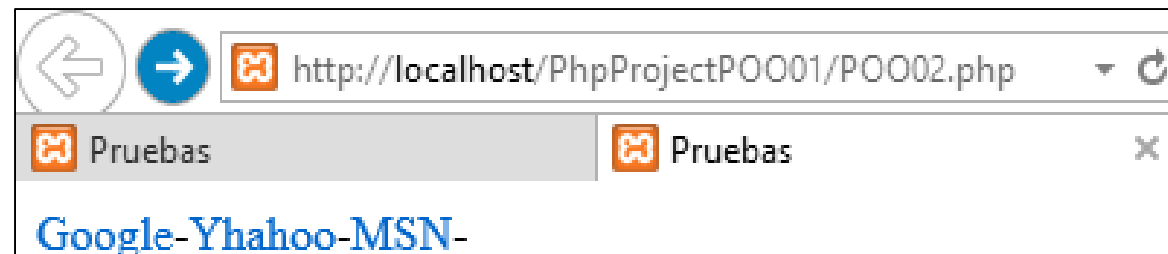
```
45 <?php
46 class Menu {
47     //Atributos de clase
48     private $enlaces=array();
49     private $titulos=array();
50
51     //Metodos de clase
52     public function cargarOpcion($en,$tit)
53     {
54         $this->enlaces[]=$en;
55         $this->titulos[]=$tit;
56     }
57     public function mostrar()
58     {
59         for($f=0;$f<count($this->enlaces);$f++)
60         {
61             echo '<a href="'. $this->enlaces[$f].'">'. $this->titulos[$f]. '</a>';
62             echo " - ";
63         }
64     }
65 }
66
67 //Creacion y asignacion de objetos
68 $menu1=new Menu();
69 $menu1->cargarOpcion('http://www.google.com','Google');
70 $menu1->cargarOpcion('http://www.yahoo.com','Yhahoo');
71 $menu1->cargarOpcion('http://www.msn.com','MSN');
72 $menu1->mostrar();
73 ?>
```

4. EJEMPLO “NAVBAR”

```
45 <?php
46 class Menu {
47     //Atributos de clase
48     private $enlaces=array();
49     private $titulos=array();
50
51     //Metodos de clase
52     public function cargarOpcion($en,$tit)
53     {
54         $this->enlaces[]=$en;
55         $this->titulos[]=$tit;
56     }
57     public function mostrar()
58     {
59         for($f=0;$f<count($this->enlaces);$f++)
60         {
61             echo '<a href="'. $this->enlaces[$f].' ">'. $this->titulos[$f]. '</a>';
62             echo " - ";
63         }
64     }
65 }
```

4. EJEMPLO “NAVBAR”

```
66 //Creacion y asignacion de objetos
67 $menu1=new Menu();
68 $menu1->cargarOpcion('http://www.google.com','Google');
69 $menu1->cargarOpcion('http://www.yahoo.com','Yhahoo');
70 $menu1->cargarOpcion('http://www.msn.com','MSN');
71 $menu1->mostrar();
72
73 ??
74 </body>
75 </html>
```



5. EJEMPLO HEADER ALINEADO



Implementar una classe “CabeceraPagina” que permeti mostrar un títol, e indicar si el volem alineat al centre, a l’esquerra o a la dreta.

5. EJEMPLO HEADER ALINEADO

```
79 <html>
80 <head>
81     <title>Pruebas</title>
82 </head>
83 <body>
84 <?php
85 class CabeceraPagina {
86     //Atributos de clase
87     private $titulo;
88     private $ubicacion;
89     //Metodos de clase
90     public function inicializar($tit,$ubi)
91     {
92         $this->titulo=$tit;
93         $this->ubicacion=$ubi;
94     }
95     public function graficar()
96     {
97         echo '<div style="font-size:40px;text-align:'.$this->ubicacion.'">';
98         echo $this->titulo;
99         echo '</div>';
100     }
101 }
102 //Creacion y asignacion de objetos
103 $cabecera=new CabeceraPagina();
104 $cabecera->inicializar('El blog del programador','center');
105 $cabecera->graficar();
106 ?>
107 </body>
108 </html>
```

5. EJEMPLO HEADER ALINEADO

```
79 <html>
80 <head>
81     <title>Pruebas</title>
82 </head>
83 <body>
84 <?php
85 class CabeceraPagina {
86     //Atributos de clase
87     private $titulo;
88     private $ubicacion;
89     //Metodos de clase
90     public function inicializar($tit,$ubi)
91     {
92         $this->titulo=$tit;
93         $this->ubicacion=$ubi;
94     }
95     public function graficar()
96     {
97         echo '<div style="font-size:40px;text-align:'. $this->ubicacion.' ">';
98         echo $this->titulo;
99         echo '</div>';
100     }
101 }
```

5. EJEMPLO HEADER ALINEADO



```
102      //Creacion y asignacion de objetos
103      $cabecera=new CabeceraPagina();
104      $cabecera->inicializar('El blog del programador','center');
105      $cabecera->graficar();
106      ?>
107      </body>
108  </html>
```


6. EJEMPLO HEADER ALINEADO

Con constructor

- Un mètode típic a les classes és el **constructor**, que seria el mètode per defecte encarregat de definir e inicialitzar un objecte d'aquesta classe.
- Aquí ho aplicarem a l'exemple anterior.

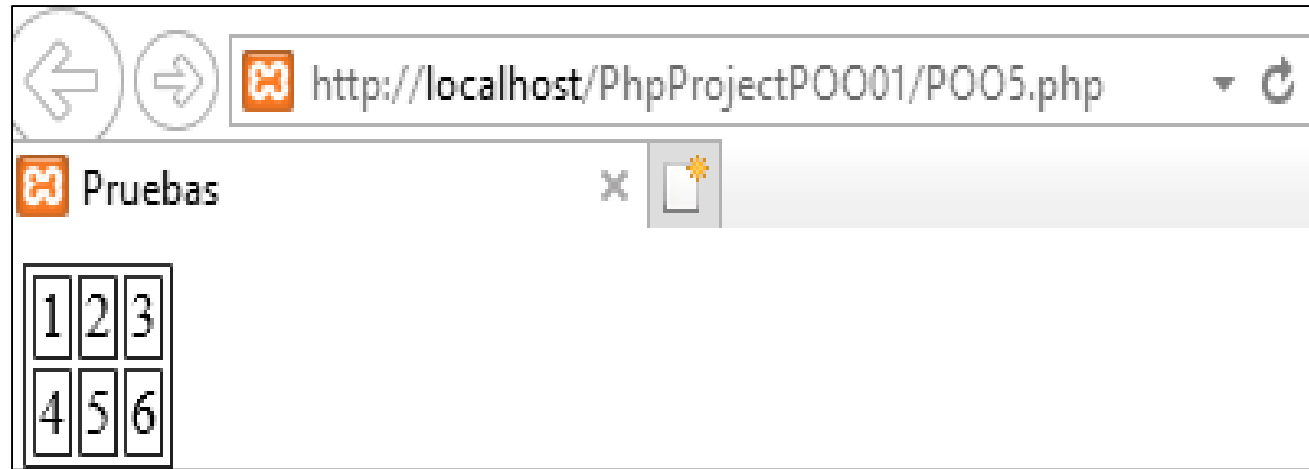


6. EJEMPLO HEADER ALINEADO

Con constructor

```
113 <?php
114 class CabeceraPagina {
115     //Atributos de clase
116     private $titulo;
117     private $ubicacion;
118     //Metodos de clase
119     public function __construct($tit,$ubi)
120     {
121         $this->titulo=$tit;
122         $this->ubicacion=$ubi;
123     }
124     public function graficar()
125     {
126         echo '<div style="font-size:40px;text-align:'.$this->ubicacion.'">';
127         echo $this->titulo;
128         echo '</div>';
129     }
130 }
131 //Creacion y asignacion de objetos
132 $cabecera=new CabeceraPagina('El blog del programador','center');
133 $cabecera->graficar();
134 ?>
```

7. EJEMPLO “TABLA”



Implementar una classe Taula que permeti indicar en el constructor la quantitat de files i columnes. Definir altre mètode que ens deixi carregar una dada concreta en una determinada cel.la. Finalment, mostrar el resultat per pantalla.

7. EJEMPLO “TABLA”

```
219 <html>
220   <head>
221     <title>Pruebas</title>
222   </head>
223   <body>
224     <?php
225       //definicion clase objeto
226       class Tabla {
227
228         //Atributos de clase
229         private $mat=array();
230         private $cantFilas;
231         private $cantColumnas;
232
233         //Metodos de clase
234         public function __construct($fi,$co)
235         {
236             $this->cantFilas=$fi;
237             $this->cantColumnas=$co;
238         }
239
```

7. EJEMPLO “TABLA”

```
233 //Metodos de clase
234 public function __construct($fi,$co)
235 {
236     $this->cantFilas=$fi;
237     $this->cantColumnas=$co;
238 }
239
240 public function cargar($fila,$columna,$valor)
241 {
242     $this->mat[$fila][$columna]=$valor;
243 }
244
245 private function inicioTabla()
246 {
247     echo '<table border="1">';
248 }
249
250 private function inicioFila()
251 {
252     echo '<tr>';
253 }
```

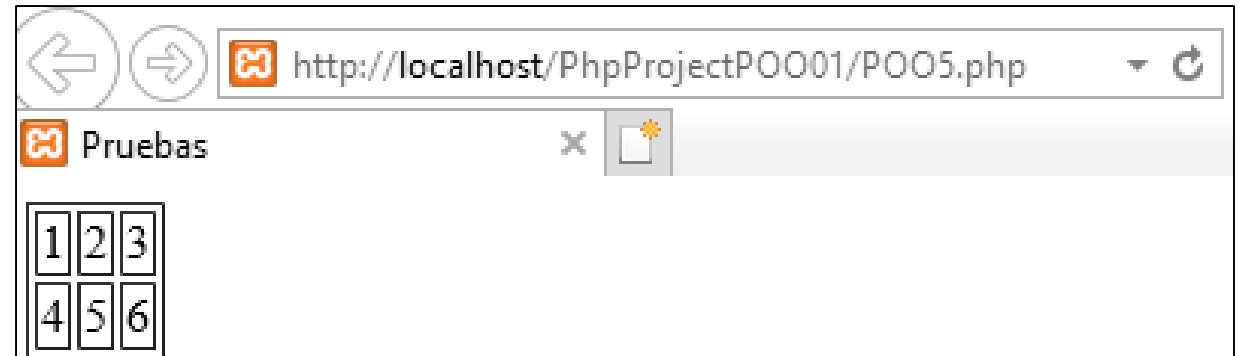
7. EJEMPLO “TABLA”

```
254
255     private function mostrar($fi,$co)
256     {
257         echo '<td>'.$this->mat[$fi][$co].'\</td>';
258     }
259
260     private function finFila()
261     {
262         echo '</tr>';
263     }
264
265     private function finTabla()
266     {
267         echo '</table>';
268     }
269
```

7. EJEMPLO “TABLA”

```
270 public function graficar()  
271 {  
272     $this->inicioTabla();  
273     for($f=1;$f<=$this->cantFilas;$f++)  
274     {  
275         $this->inicioFila();  
276         for($c=1;$c<=$this->cantColumnas;$c++)  
277         {  
278             $this->mostrar($f,$c);  
279         }  
280         $this->finFila();  
281     }  
282     $this->finTabla();  
283 }  
284 }
```

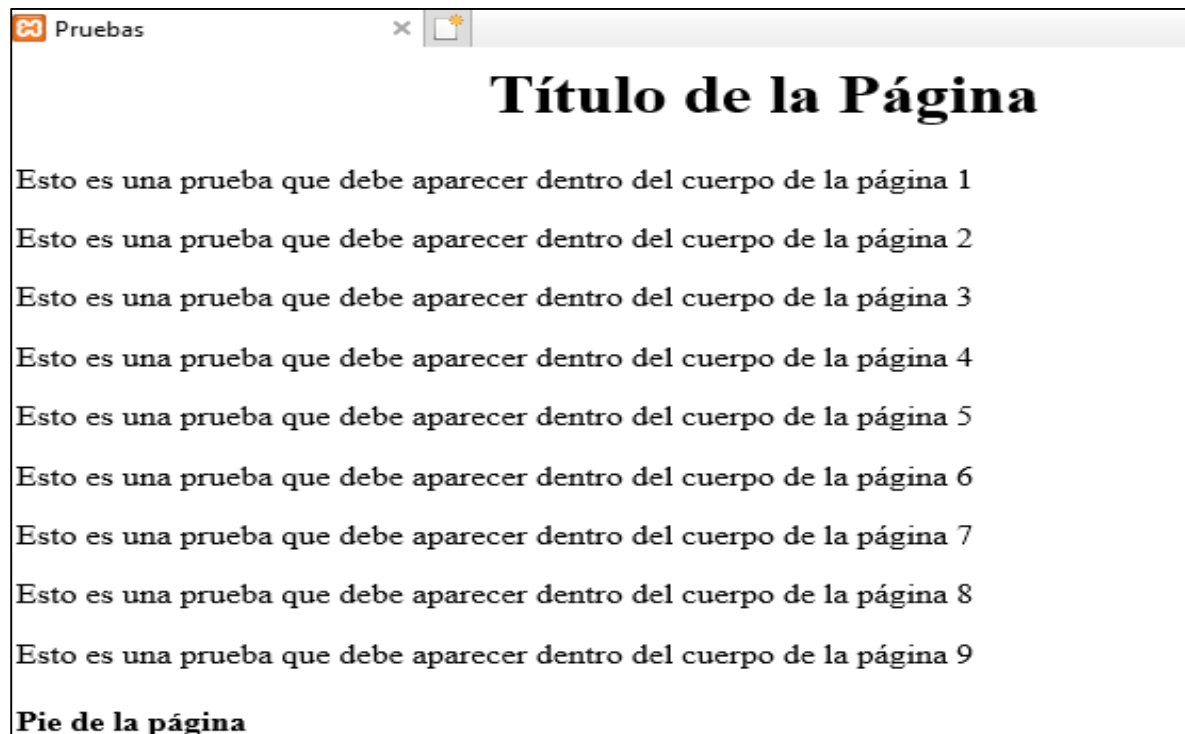
7. EJEMPLO "TABLA"



```
285 //Creacion y asignacion de objetos
286 $tablal=new Tabla(2,3);
287 $tablal->cargar(1,1,"1");
288 $tablal->cargar(1,2,"2");
289 $tablal->cargar(1,3,"3");
290 $tablal->cargar(2,1,"4");
291 $tablal->cargar(2,2,"5");
292 $tablal->cargar(2,3,"6");
293 $tablal->graficar();
294 ?>
295 </body>
296 </html>
```


8. EJEMPLO “PAGINA”

Els objectes poden establir relacions entre si. De fet, de vegades és imprescindible per a tenir una bona solució a un problema concret.



Plantejar una classe Pagina que inclogui objectes de la classe Capçalera, Cos i Peu. Capçalera i Peu han de tenir un atribut on s'enmagatzemi el text a mostrar. La classe Cos ha de tenir un atributo tipus *array* on s'enmagatzemin tots els paràgrafs.

8. EJEMPLO "PAGINA"

```
79 <html>
80 <head>
81     <title>Pruebas</title>
82 </head>
83 <body>
84 <?php
85 class CabeceraPagina {
86     //Atributos de clase
87     private $titulo;
88     private $ubicacion;
89     //Metodos de clase
90     public function inicializar($tit,$ubi)
91     {
92         $this->titulo=$tit;
93         $this->ubicacion=$ubi;
94     }
95     public function graficar()
96     {
97         echo '<div style="font-size:40px;text-align:'. $this->ubicacion. '">';
98         echo $this->titulo;
99         echo '</div>';
100     }
101 }
```

8. EJEMPLO “PAGINA”

```
322 //definicion clase objeto
323 class Cuerpo {
324     //Atributos de objeto
325     private $lineas=array();
326     //Metodos de objeto
327     public function insertarParrafo($li)
328     {
329         $this->lineas[]=$li;
330     }
331     public function graficar()
332     {
333         for($f=0;$f<count($this->lineas);$f++)
334         {
335             echo '<p>'.$this->lineas[$f].'</p>';
336         }
337     }
338 }
```

8. EJEMPLO “PAGINA”

```
339 //definicion clase objeto
340 class Pie {
341     //Atributos de objeto
342     private $titulo;
343     //Constructor de objeto
344     public function __construct($tit)
345     {
346         $this->titulo=$tit;
347     }
348     //Metodos de objeto
349     public function graficar()
350     {
351         echo '<h4 style="text-align:left">'.$this->titulo.'</h4>';
352     }
353 }
```

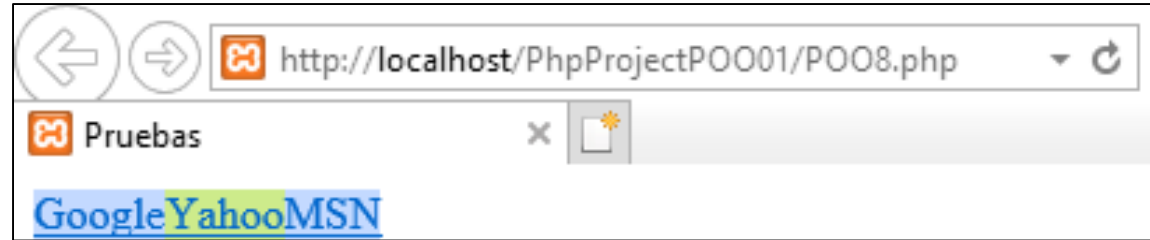
8. EJEMPLO “PAGINA”

```
354 //definicion clase objeto
355 class Pagina {
356     //Atributos de objeto
357     private $cabecera;
358     private $cuerpo;
359     private $pie;
360     //Constructor de objeto
361     public function __construct($texto1,$texto2)
362     {
363         $this->cabecera=new Cabecera($texto1);
364         $this->cuerpo=new Cuerpo();
365         $this->pie=new Pie($texto2);
366     }
367     //Metodos de objeto
368     public function insertarCuerpo($texto)
369     {
370         $this->cuerpo->insertarParrafo($texto);
371     }
372     public function graficar()
373     {
374         $this->cabecera->graficar();
375         $this->cuerpo->graficar();
376         $this->pie->graficar();
377     }
378 }
```

8. EJEMPLO “PAGINA”

```
380 //Creacion y asignacion de objetos
381 $paginal=new Pagina('Título de la Página','Pie de la página');
382 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 1');
383 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 2');
384 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 3');
385 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 4');
386 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 5');
387 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 6');
388 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 7');
389 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 8');
390 $paginal->insertarCuerpo('Esto es una prueba que debe aparecer dentro del cuerpo de la página 9');
391 $paginal->graficar();
392 ?>
393 </body>
394 </html>
```

9. EJEMPLO “OBJETOS A UN MENU”



- Plantejarem una classe Opcion i una altra classe Menu. La classe Opcion definirà com atribut el títol, l'enllaç i el color de fons. Els mètodes a implementar seran el constructor i el renderitzador/graficador.
- Per altra banda, la classe Menú administrará un *array* d'objectes de la classe Opcion i implementarà mètodes per a inserir objectes Opcion a la classe Menu i un altre per a renderitzar. A més, al constructor de la classe Menú l'indicarem si volem orientació horitzonal o vertical.

9. EJEMPLO “OBJETOS A UN MENU”

```
399 <html>
400 <head>
401     <title>Pruebas</title>
402 </head>
403 <body>
404 <?php
405 //definicion clase objeto
406 class Opcion {
407     //Atributos de objeto
408     private $titulo;
409     private $enlace;
410     private $colorFondo;
411     //Constructor de objeto
412     public function __construct($tit,$enl,$cfon)
413     {
414         $this->titulo=$tit;
415         $this->enlace=$enl;
416         $this->colorFondo=$cfon;
417     }
418     //Metodos de objeto
419     public function graficar()
420     {
421         echo '<a style="background-color:'. $this->colorFondo.'" href="'. $this->enlace.'">'. $this->titulo.'</a>';
422     }
423 }
```

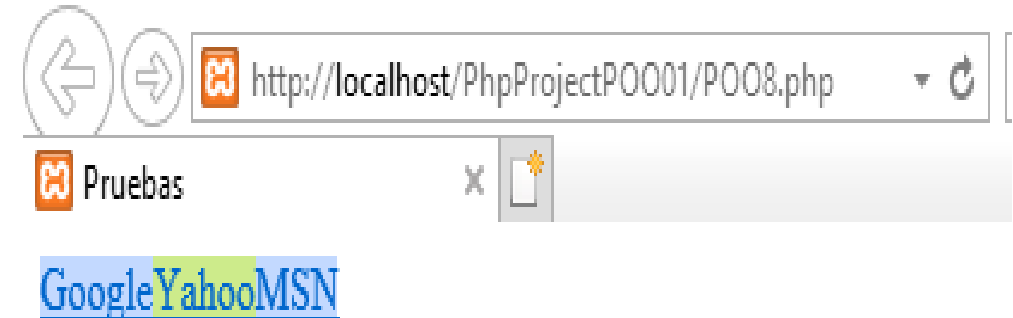

9. EJEMPLO "OBJETOS A UN MENU"

```
424 //definicion clase objeto
425 class Menu {
426     //Atributos de objeto
427     private $opciones=array();
428     private $direccion;
429     //Constructor de objeto
430     public function __construct($dir)
431     {
432         $this->direccion=$dir;
433     }
434     //Metodos de objeto
435     public function insertar($op)
436     {
437         $this->opciones[]=$op;
438     }
439
440     private function graficarHorizontal()
441     {
442         for($f=0;$f<count($this->opciones);$f++)
443         {
444             $this->opciones[$f]->graficar();
445         }
446     }
```

9. EJEMPLO “OBJETOS A UN MENU”

```
447     private function graficarVertical()  
448     {  
449         for($f=0;$f<count($this->opciones);$f++)  
450         {  
451             $this->opciones[$f]->graficar();  
452             echo '<br>';  
453         }  
454     }  
455     public function graficar()  
456     {  
457         if (strtolower($this->direccion)=="horizontal")  
458             $this->graficarHorizontal();  
459         else  
460             if (strtolower($this->direccion)=="vertical")  
461                 $this->graficarVertical();  
462     }  
463 }
```

9. EJEMPLO “OBJETOS A UN MENU”



```
464 //Creacion y asignacion de objetos
465 $menu1=new Menu('horizontal');
466 $opcion1=new Opcion('Google','http://www.google.com','#C3D9FF');
467 $menu1->insertar($opcion1);
468 $opcion2=new Opcion('Yahoo','http://www.yahoo.com','#CDEB8B');
469 $menu1->insertar($opcion2);
470 $opcion3=new Opcion('MSN','http://www.msn.com','#C3D9FF');
471 $menu1->insertar($opcion3);
472 $menu1->graficar();
473
474 ?>
475 </body>
476 </html>
```

10. EJEMPLO “HEADER CONFIGURADO”



- Desenvolupar la classe CabeceraDePagina, que ens mostrarà un títol alineat amb un determinat color de Font i de fons.
- S’han de definir paràmetres opcionals tant pels colors com per l’alineació del títol.

10. EJEMPLO “HEADER CONFIGURADO”

```
487 <?php
488 //definicion clase objeto
489 class CabeceraPagina {
490     //Atributos de objeto
491     private $titulo;
492     private $ubicacion;
493     private $colorFuente;
494     private $colorFondo;
495     //Constructor de objeto
496     public function __construct($tit,$ubi='center',$colorFuen='#ffffff',$colorFon='#000000')
497     {
498         $this->titulo=$tit;
499         $this->ubicacion=$ubi;
500         $this->colorFuente=$colorFuen;
501         $this->colorFondo=$colorFon;
502     }
503     //Metodos de objeto
504     public function graficar()
505     {
506         echo '<div style="font-size:40px;text-align:'.$this->ubicacion.';color:';
507         echo $this->colorFuente.';background-color:'.$this->colorFondo.'">';
508         echo $this->titulo;
509         echo '</div>';
510     }
511 }
```

10. EJEMPLO “HEADER CONFIGURADO”

```
512 //Creacion y asignacion de objetos
513 $cabecera1=new CabeceraPagina('El blog del programador');
514 $cabecera1->graficar();
515 echo '<br>';
516 $cabecera2=new CabeceraPagina('El blog del programador','left');
517 $cabecera2->graficar();
518 echo '<br>';
519 $cabecera3=new CabeceraPagina('El blog del programador','right','#ff0000');
520 $cabecera3->graficar();
521 echo '<br>';
522 $cabecera4=new CabeceraPagina('El blog del programador','right','#ff0000','#ffff00');
523 $cabecera4->graficar();
524
525 ?>
526 </body>
527 </html>
```